Ring Road NoC Architecture

Henrik Samuelsson and Shashi Kumar

School of Engineering, Jönköping University,
Department of Electronic/Computer Engineering,
P.O. Box 1026, S-551 11 Jönköping, Sweden

{ef02sahe, kush}@ing.hj.se

Abstract

A network of routers on a chip which allows packet switched communication, referred as Networks on Chip (NoC), provides a scalable interconnection architectural option for rapid development of SoCs. In this paper we describe a new NoC architecture, called Ring Road NoC (R^2NoC), which was motivated by smooth flow of traffic in interconnected ring roads. Router design and routing algorithm for R^2NoC is much simpler. We have developed a SystemC model of R^2NoC and our simulation studies demonstrate that the proposed architecture leads to smooth and evenly distributed flow of communication traffic.

1 Introduction

Network on Chip (NoC) is emerging as a new paradigm for solving interconnection problems in the design of core based Systems on Chip (SoCs). In this paradigm the cores are connected through on-chip routers and send data to each other through packet switched communication. The on-chip communication infrastructure is reusable across systems. Therefore, it reduces time to design. Many researchers have proposed routing architectures which are scalable [1 ,2].

Among the different NoC proposals that have been proposed by researchers around the world are the following. Royal Institute of Technology (KTH) have research in the NoC area, their concept is named Nostrum [3]. The Nostrum Mesh Architecture is based on a 2d-mesh structure where each Router is connected to one resource and four other routers. The addressing scheme used is based on Cartesian coordinates. Philip's Research Laboratories has implemented a NoC named Æthereal [4]. The construction is topology independent, due to an addressing scheme based on source routing. Two traffic classes are supported guaranteed throughput data (GT) and best-effort data (BT). A circuit is set up for the GT traffic based on time division; this gives quality of service for some communicating pairs of resources. Tampere University of Technology in Finland has developed a NoC based on a heterogeneous network named Proteo [5]. A Proteo based network is built up from library of communication blocks. A typical Proteo NoC is divided in clusters in a hierarchical style. The different part of design can follow completely different standards regarding the traffic. Some part of the network may include support for quality of service, while another subnet may not.

Although offering many advantages, NoC paradigm introduces new problems. Due to packet switched nature of on-chip communication there may be different amount of traffic in different parts of the network. There may be congestion in routers in certain paths and some routers and links may be highly under-utilized. In certain situations packets may be lost or dropped before they reach the destination. Guaranteeing latency and throughput in packet switched network is also harder than in circuit switched networks. The on-chip routers also contribute to design and cost overheads in a NoC based design. In this paper, we describe a novel NoC architecture called Ring Road NoC (R^2NoC) which has potential of offering smoother on-chip communication.

2 R^2NoC – basic ideas

A new NoC architecture is proposed in this paper. The topology is inspired by ring roads used in big cities. The idea of ring roads is to lead some of the traffic around the central part of a city. Some traffic is forced to take a detour around the center. The use of this strategy reduces the risk of congestion in the central parts of the network.

2.1 Motivation for a new topology

An often proposed NoC architecture is based on the Manhattan-style mesh topology. There are several benefits with this choice of topology. The structure is good from a layout perspective. The network scales up in a natural way as the number of resources grow, without need for redesign of switches and resources. The addressing scheme is straightforward and the routing becomes simple, this gives lightweight routers that can operate at high clock speed with low overhead. A drawback of the Manhattan-style mesh topology is that the load on the routers becomes uneven. The traffic load in the central parts of the network becomes higher than the load in the periphery parts. This problem becomes more significant as the network size scales up.

The objective is to find an architecture that spread the traffic as evenly as possible without complicated routers.

2.2 Basic structure

A R^2NoC has a topology with a ring structure, these rings lead some traffic around the center of the network. The traffic is packet based. A packet that is sent between two resources located on the same ring will never leave the ring. This strategy simplifies the routing and gives even spread of the traffic over the entire network. A drawback is that some packets will travel a bit longer than necessary since the shortest path is sometimes through the center of the network.

An example of one possible R^{2}NoC is depicted in Figure 1. There is 36 resources connected to three rings in this example. The resources in the network are the bigger blocks with indentations on the side that mark which side of the block that is connected to the network. The smaller blocks are the routers in the network. The numbers represents the addresses of the routers. The rings are connected with links that allow packets to switch rings. These links are only used if the sending and receiving resource are located on different rings. The R^{2}NoC presented in Figure 1 has three rings. It is straightforward to scale up the system by adding more rings.

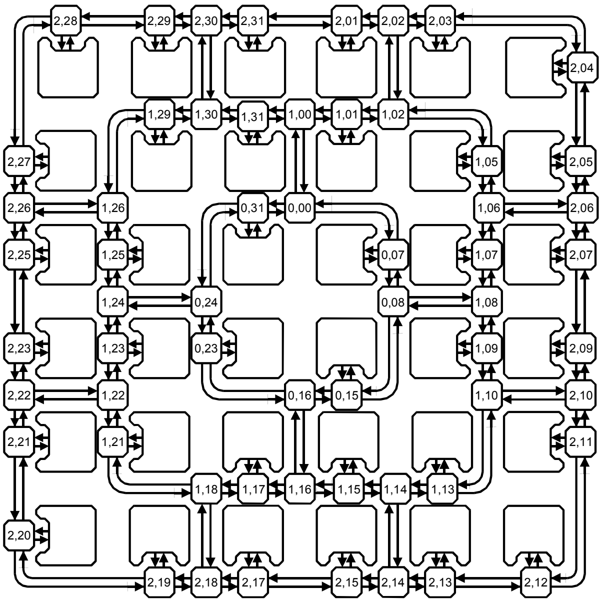

Figure 1. A possible R^{2}NoC topology.

3 R^{2}NoC – architectural details

3.1 Router types

There are two types of routers in a R^{2}NoC; the first type connects the resources to the network and is hereby named Network-Resource Router (NRR), the second kind connects the rings of the network and comes in two subtypes named Ring Router type 1 (RR1) and type 2 (RR2). It is the Ring Routers that makes it possible for a packet to leave the current ring and enter a new ring. The Ring Routers always come in pairs. RR1 of a pair enables packets to move to an outer ring and RR2 enables packets to enter an inner ring.

3.2 Addressing

The addressing of the resources and routers used in a R^{2}NoC is based on a variant of polar coordinates. Polar coordinates are an option that simplifies life when circular symmetry is present. The location of a point is described by its distance from the origin called r and the angle θ between the line from the origin and the reference axis which is usually the X-axis. It is however not possible to directly use addressing according to polar coordinates in the R^{2}NoC. The reason for this is that the ring roads in the R^{2}NoC are not circle shaped; instead rings are almost square-like because of layout reasons. It is desirable to have square shaped spaces of equal size as base for the resource placement.

The addresses are on the form (r, θ). The numbers in Figure 1 represent the addresses used in this example R^{2}NoC. The r component is simply the number of the ring that the resource or router is placed in. All the resources and routers that are connected to the innermost ring have r set to 0. The second smallest ring has r set to 1 and then r grows correspondingly as the radius of the rings increases. The θ part of the address is slightly more complicated. θ is based on the angle relative to a reference axle like in ordinary polar coordinates. The reference axle is chosen to be a an imaginary vertical line starting in the center of the network and pointing upwards. Routers and resources that are located on this line all have θ part of the address set equal to zero. θ will then increase as a given ring is traversed clockwise. The number of different θ that are needed depends on the size and the router placement of the network. Study Figure 1 again, this R^{2}NoC uses θ values in the range of 0 to 31.

3.3 Routing in R^{2}NoC

The address information of the receiving resource is stored on the (r, θ)-format in every packet that is sent out from the sending resource to the network. The behavior of the routers depends on this address information and is accounted for below.

Each router has three input/output pairs that are numbered 1, 2 and 3 according to figure 2.

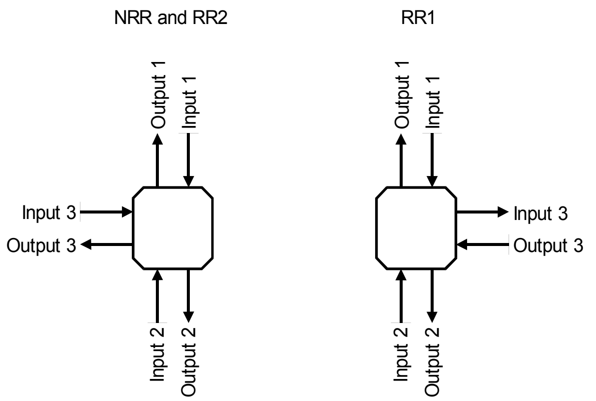

Figure 2. Naming convention used for the router ports.

We start by studying packets that arrive through input 3. Packets that enters the routers this way comes either directly from a resource or from another ring in the network. The packet will be sent on to Output 1 or 2 depending on a comparison of the θ component in the packet and θ component of the router. The packet will be sent on either counter clockwise through Output 1 or clockwise through Output 2 depending on which way is shortest for the packet to take when traversing the network to the destination resource.

The routing of packets that enter Input 1 or Input 2 of another router need to be divided in to three sub cases depending of the type of the receiving router. Assume first that the router is of type RR1, a comparison of the r components is performed to decide whether to let the packet continue to travel on the same ring or move the packet out to the next outer ring. The packet will be sent to an outer ring through Output 3 if $r_{packet} > r_{router}$, otherwise will the packet continue to travel on the same ring in the same direction by leaving the router through Output 1 or 2. Secondly the router may be of type RR2, the routing in this case is analogous to the first case. The only difference is that the packet may be sent to an outer ring. In the third case the packet enters a NRR and a check is performed of both the r and θ components to check if the packet has arrived at its destination. The packet is sent on through output 3 to the connected resource if both components match, otherwise will the packet continue to traverse the network.

The different routing decisions accounted for above vary in complexity. The most complex case is when a packet enters Input 3 of a router. The routing decision is based on a comparison of the θ variable that is stored in a number of bits that has to be compared. This introduces the biggest delay and hardware costs. The second most complex case is when a packet enters through Input 1 or 2 of a RR when a comparison of the r variable that usually consists of a fewer number of bits shall be compared. A shortcut can also be used in this case, since if the r values match so can the packet be sent on directly to the opposite output without the time consuming comparison. The simplest case is when a packet enters Input 1or 2 of a NRR where only a very simple check has to be performed to see if both the address variables matches. Note that the less advanced routing decisions are the most common which give low total delay assuming that a asynchronous design style i employed.

3.4 Architectural variations

The architecture proposed in Figure 1 may not be the ultimate solution. A factor that can be varied is the ratio of the different routers in the architecture. In the first introduced examples above every second router in a ring is of type RR, except for the outer ring that makes up a special border case. The performance may for example improve if every third or fourth router is a RR. It is possible to move more resources to the central rings if the removed RRs are replaced with NRRs. It is in this way possible to have a R^2NoC with equal number of resources on each ring which may be desirable in some situations.

Heterogeneous resource sizes can also be supported since the the removal of one or more NRR and its corresponding resource makes it possible to increase the size of the remaining neighboring resources. Further variations and details are discussed in [8].

4 Modeling and Performance Analysis

A model of an R^2NoC has been constructed to evaluate the functionality and performance. The topology and size of the model is chosen the one in Figure 1 with three rings and 36 resources that was presented in section 2.2.

4.1 SystemC Model of R^2NoC

A model of a system enables designers to test the functionality and experiment with different parameter settings before the a hardware version is implemented. The modeling language chosen for this project is SystemC [6], other possible options are for example SDL discussed in [7]. SystemC models are built up by modules that are connected together with signals. There is one type of resource modules and three types of router modules in the R^2NoC model. The behavior of resource module is implemented by two processes that run concurrently. One process generates the traffic in the form of packets that are sent out on the network. The other process takes care of the packets sent to the resource. The routers are built up in a hierarchical way with submodules as illustrated by Figure 3.

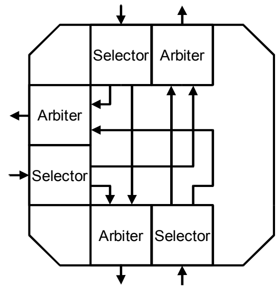

Figure 3. Router buildup in the SystemC model.

The routers relies on a store and forward technique. Each input of the routers has a selector module that performs the actual routing by selecting to send the packet on to one of two possible outputs. One Arbiter module is located on each output, the arbiters make sure that only one Selector is allowed to send packets at a given time. The Selector modules are served by the arbiters in a Round Robbin fashion. Another duty of the arbiter modules is to communicate with the receiving router/resource to make sure that it is ready to accept a new packet.

4.2 Traffic Modeling

Two different types of traffic can be generated by the resources, totally random traffic and biased random traffic. When the traffic generation is set to random traffic so will the address of the receiving resource be set to a totally random address regardless of the address of the sending resource. Biased traffic means that it is more likely that the receiving resource will be one that lies close to the sending resource.

4.4 Results

Various simulations have been run using the model to test the R^2NoC ideas. One of the main ideas of the R^2NoC is that the topology shall give even distribution of the traffic over the entire network. It is therefore interesting to measure how the stress varies over the network. This can be done by logging how many packets the different parts of the network will handle during a simulation. All the measurements in this section has been done with low load on the network, since the goal for now is to determine how the traffic spreads not how much stress the network can take.

The first class of simulation runs uses random traffic where the traffic is spread evenly over the network regardless of the address of the sending resource. The result can be seen in Figure 4 and are analysed below.

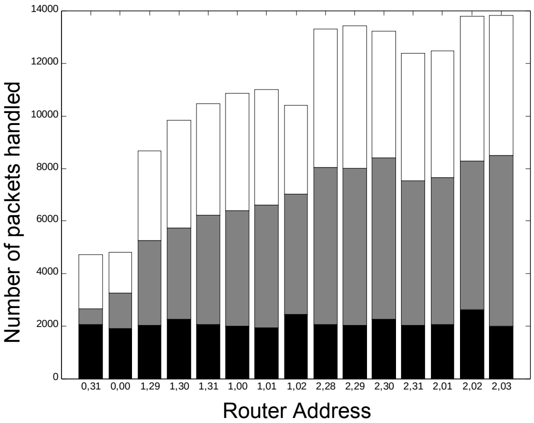

Figure 4. Traffic spread with random traffic.

Symmetry of the network means that only one fourth of the network need to be analysed since the behaviour will be similar in the four parts, the north part of the network is presented here, the addresses of the routers can be seen in the bottom of the diagrams, refer to Figure 1 for their actual placement in the network. The bars represents the total number of packets handled by each studied router during the simulation. The white parts in the bars represents the number of packets that enter the routers through the Input 1, the grey part are packet that enter the router through Input 2. Finally black represents packet that enters via Input 3, that is packets which come from either a resource or another ring. There is most stress on the router in the outer ring in this case. The middle ring has a little less stress and finally the innermost ring handles very few packets. The routers of the outer ring must handle about three times as many packets as the innermost ring. The remedy for this problem is to move some more of the resources to the inner rings if this kind of traffic is present in the application.

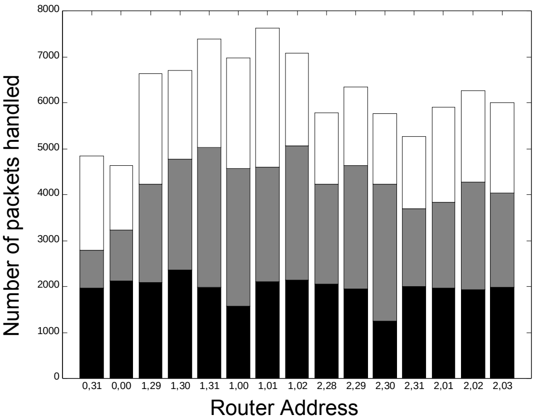

Figure 5. Traffic spread with biased traffic.

The second class of simulation that has been run uses the biased traffic when it is more likely that the receiving resource is one close to the sender according to a Poisson distribution, the results are presented in Figure 5. It turns out that the load on the routers becomes more even in this case. The amount of handled packets vary by a factor less than two which is good since it means that no part of the network is greatly under- or over-utilized. The reason for this is related to the behaviour of the resources placed on the boundary. They do not have so many neighbours to send to so they will more often send packets inwards to the centre increasing the load in the middle of the network. More results and details can be found in [8].

5 Conclusion

We have described a novel NoC architecture which has many desirable properties, like simple router design, simple routing algorithm, easy 2D layout, small risk of congestion et cetera and a potential for good performance. A lot of architectural variations are possible in the choice of numberof rings, ratio between various types of routers, application (traffic) dependent routing algorithms etc. Many of these issues could be used for interesting future research.

Acknowledgements

The authors would like to thank Rickard Holsmark for many usefuel discussions during developement of R^2NoC architecture. The research reported in this paper was carried under the reserach project ”Specification and Evaluation of Network on chip architectures for multimedia applications”, founded by Swedish K. K. Foundation.

References

[1] Axel Jantsch, *Network on chip,* Proceedings of the Conference Radio Science and communication, Stockholm (2002).

[2] W. J. Dally and B. Towles, *Route Packets, Not Wires: On-Chip Interconnection Networks,* Design Automation Conference, 2001.

[3] Mikael Millberg et al.,*Guaranteed Bandwidth using Looped Containers in Temporally Disjoint Networks within the Nostrum Network on Chip*, Royal Institute of Technology (KTH), 2004.

[4] John Dielissen et al., *Concepts and Implementation of the Philips Network-on-Chip*, Philips Research Laboratories, 2003.

[5] Sigüenza-Tortosa, D., Nurmi, J.,*Proteo: A New Approach to Network-on-Chip.,* Proceedings of the IASTED International Conference on Communication Systems and Networks, 2002, pp. 355-357.

[6] *SystemC 2.0.1 Language Reference Manual*, San Jose, 2003.

[7] R. Holsmark et al. *Modelling and Evaluation of a Network on Chip Architecture using SDL*, 11th SDL Forum, SDL 2003, Germany, 1-14 July 2003.

[8] H. Samuelsson, *Ring Road Network on Chip Architecture*, master of science thesis, School of Engineering, Jönköping University, 2004.